

Travaux pratiques 3 Programmation orientée objet (Java) SMA5

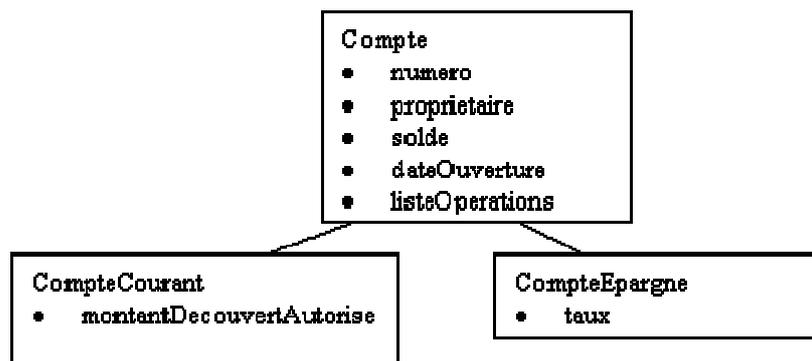
Exercice 1: Gestion des comptes bancaires

Cet exercice a pour but la gestion des comptes d'une banque.

1. Une opération bancaire est définie par le montant de l'opération (positif ou négatif), la date de cette opération et un libellé. (**Exemple** : 19/1/2004 | cheque n° 123456 | -53,60 DH)

Ecrire une classe **Operation** permettant de modéliser de telles opérations : définir les attributs nécessaires, qui seront encapsulés ; écrire un constructeur ; les accesseurs (méthodes get...), et redéfinir la méthode **toString**

2. Simplifier les types de comptes bancaires en considérant le diagramme de classes suivant :



- Ecrire chacune de ces classes. Choisir avec perspicacité le type de chaque attribut, en expliquant plus particulièrement le choix de la structure pour représenter sous la forme d'une liste l'historique des opérations effectuées sur un compte.
- Ecrire pour chaque classe un ou plusieurs constructeurs appropriés.
- Encapsuler chaque attribut, et écrire une méthode **get_** correspondant à chaque attribut.
- Redéfinir la méthode **toString** de chaque classe permettant de construire une String représentant les informations caractérisant un de ces comptes bancaires.

3. Dans la classe **Compte**, écrire 2 méthodes **verser(float montant, String libelle)** et **retirer(float montant, String libelle)** permettant de verser et de retirer de l'argent sur le compte, en créant à chaque fois un objet de la classe **Operation** qui sera rangé dans la liste des opérations, afin d'en garder un historique.

Ecrire une classe **EMontantMinException**, héritant de la classe **Exception** qui pourra être déclenchée par la méthode retirer si on essaye de retirer plus d'argent que le solde du compte ou le montant du découvert autorisé.

NB : la méthode retirer devra être définie au niveau de la classe **Compte** et redéfinie au niveau de la classe **CompteCourant** pour traiter le cas du découvert autorisé.

4. Ecrire une méthode **float CalculInterets** dans la classe **CompteEpargne** qui calcule les intérêts produits par ce compte (résultat de la fonction). Ces intérêts seront versés sur le compte.

5. Ecrire une méthode **Historique(int nombreOperations)** affichant le solde du compte et la succession des dernières "*nombreOperations*" opérations.

6. Ecrire un programme Java pour utiliser ces classes. Ce programme proposera tout d'abord de créer un compte en demandant s'il s'agit d'un compte courant ou d'un compte épargne.

Ensuite il proposera un système de choix permettant d'effectuer une succession d'opérations sur ce compte : versement, retrait, calcul et ajout des intérêts (uniquement pour les comptes épargnes), consultation du solde et de l'historique.

Exercice 2: Tours de Hanoi

Le but de cet exercice est d'écrire **le jeu de la tour de Hanoi** avec des fonctions récursives. Dans un premier temps, on manipulera un jeu avec 5 anneaux. Une fois l'exercice terminé, on pourra essayer avec plus d'anneaux.

1. Déclarer une classe qui représente une pile d'entiers et écrire les méthodes qui permettent d'empiler et de dépiler un entier. Ecrire également une méthode qui affiche le contenu de la pile.

2. Déclarer une classe **Hanoi** qui contient un tableau de trois piles qui représentent les trois poteaux du jeu, et écrire le constructeur de la classe qui initialise le jeu en empilant les 5 anneaux correctement sur l'un des trois poteaux. Ecrire également une méthode qui affiche à l'écran l'état du jeu à un instant donné.

3. Ecrire une méthode qui déplace les 5 anneaux sur l'un des deux poteaux libres en respectant les règles du jeu. Faire en sorte que l'affichage à l'écran montre bien le déplacement de chaque anneau.

4. Reprendre les classes **Pile** et **Hanoi**, et écrire une applet qui anime le déplacement des anneaux.